UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/932,578	08/17/2001	Blake Lewis	103.1072.01	5197

48102 7590 02/07/2007

NETWORK APPLIANCE/BLAKELY
12400 WILSHIRE BLVD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
LE, MIRANDA	
ART UNIT	**PAPER NUMBER**
2167	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	09/932,578	LEWIS ET AL.	
	Examiner	Art Unit	
	Miranda Le	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 November 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *25-35 and 40-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *25-35 and 40-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This communication is responsive to Amendment, filed 11/17/06.

Claims 25-35, 40-48 are pending in this application. This action is made Final.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 25, 26, 28-31, 35, 43, 44, 46-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Armangau et al. (US Patent No. 6,434,681), in view of Forin et al. (US Patent No. 6,175,900).

As per claim 25, Armangau teaches a method comprising:

maintaining an active map *(i.e. data structure, col. 13, lines 25-65)* of information indicating in-use and free blocks associated with a file system *(i.e. a list of pointers to free tracks, col. 13, line 66 to col. 14, line 7);*

maintaining a set of snapshots *(i.e. snapshot copies, col. 13, lines 25-57)*, each snapshot representing a state *(i.e. the bit map indicates the modified state, col. 13, lines 58-65)* of said file system at a particular point in time *(i.e. date/time stamp when the snapshot copy was made, col. 18, lines 18-36); and*

computing a summary map *(Fig. 6)* of active maps included in at least two of said snapshots *(i.e. snapshot track 0, snapshot track 1, Fig. 6) (col. 14, line 62 to col. 15, line 18).*

Armangau does not expressly teach "as a logical union".

Forin teaches computing a summary map *(i.e. a hierarchical bit-map 124, Fig. 8a)* as a logical union *(i.e. When two or more neighboring blocks are marked as free, they are automatically and immediately available for allocation as a single piece of memory, col. 9, lines 31-41)* of active maps *(i.e. the bit map, col. 7, lines 13-18) (col. 9, lines 9-51).*

It should be noted that the two neighboring free blocks has two pairs of free two-bit values 01, 01 (See Table 1 in col. 7), then, after the step of coalescing (i.e. computing), a single piece of memory is free having a free two-bit value 01, which is as result of logical union of 01 and 01.

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau and Forin at the time the invention was mage to modify the system of Armangau to include computing a summary map as a logical union of active maps as taught by Forin. One of ordinary skill in the art would be motivated to make this combination in order to represent an allocation state of the corresponding memory block in view of Forin, as doing so would give the added benefit of managing the memory blocks and tracks changes in allocation state via hierarchical bit map as taught by Forin *(col. 4, lines 24-30).*

As per claim 35, Armangau teaches a method comprising:

maintaining an active map *(i.e. data structure, col. 13, lines 25-65)* of information indicating in-use and free blocks associated with a file system *(i.e. a list of pointers to free tracks, col. 13, line 66 to col. 14, line 7);*

maintaining a set of snapshots *(i.e. snapshot copies, col. 13, lines 25-57)*, each snapshot representing a state *(i.e. the bit map indicates the modified state, col. 13, lines 58-65)* of said file system at a particular point in time *(i.e. date/time stamp when the snapshot copy was made, col. 18, lines 18-36); and*

maintaining a summary map *(Fig. 6)* of active maps included in at least two of said snapshots *(i.e. snapshot track 0, snapshot track 1, Fig. 6) (col. 14, line 62 to col. 15, line 18);*

selecting a set of blocks *(i.e. a write operation upon a storage location, col. 2, lines 12-34)* maintained by said file system to which perform a write allocation operation *(col. 2, lines 12-65);*

updating *(i.e. placing a pointer to the allocated storage location of the snapshot copy on the list of pointers allocated to the snapshot copy, changing the bit in the bit map for the storage location of the production data set, col. 2, lines 25-65)* only a portion of said summary map corresponding to said set of blocks, in response to said selecting *(col. 2, lines 12-65); and*

performing said write allocation *(i.e. performing a write operation upon the storage location of the production data set, col. 2, lines 35-65)* operation in response to said updated summary map *(col. 2, lines 12-65).*

Armangau does not expressly teach "computed as a logical union".

Forin teaches a summary map *(i.e. a hierarchical bit-map 124, Fig. 8a)* computed as a logical union *(i.e. When two or more neighboring blocks are marked as free, they are automatically and immediately available for allocation as a single piece of memory, col. 9, lines 31-41)* of active maps *(i.e. the bit map, col. 7, lines 13-18) (col. 9, lines 9-51).*

It should be noted that the two neighboring free blocks has to two pair of free two-bit values 01, 01 (See Table 1 in col. 7), then, after the step of coalescing (i.e. computing), a single piece of memory is free having a free two-bit value 01, which is as result of logical union of 01 and 01.

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau and Forin at the time the invention was mage to modify the system of Armangau to include maintaining a summary map computed as a logical union of active maps as taught by Forin. One of ordinary skill in the art would be motivated to make this combination in order to representing an allocation state of the corresponding memory block in view of Forin, as doing so

would give the added benefit of managing the memory blocks and tracks changes in allocation state via hierarchical bit map as taught by Forin *(col. 4, lines 24-30).*

As per claim 43, Armangau teaches a method comprising:

maintaining an active map *(i.e. data structure, col. 13, lines 25-65)* of information indicating in-use and free blocks associated with a file system *(i.e. a list of pointers to free tracks, col. 13, line 66 to col. 14, line 7);*

maintaining a plurality of persistent point-in-time images *(i.e. snapshot copies, col. 13, lines 25-57; snapshot volume, Fig. 5)*, each persistent point-in-time image representing a state *(i.e. the bit map indicates the modified state, col. 13, lines 58-65)* of said file system at a particular point in time *(i.e. date/time stamp when the snapshot copy was made, col. 18, lines 18-36); and*

generating a summary map *(Fig. 6)* of active maps included in at least two of said persistent point-in-time images *(i.e. snapshot track 0, snapshot track 1, Fig. 6) (col. 14, line 62 to col. 15, line 18);*

Armangau does not expressly teach "as a logical union".

Forin teaches generating a summary map *(i.e. a hierarchical bit-map 124, Fig. 8a)* as a logical union *(i.e. When two or more neighboring blocks are marked as free, they are automatically and immediately available for allocation as a single piece of memory, col. 9, lines 31-41)* of active maps *(i.e. the bit map, col. 7, lines 13-18) (col. 9, lines 9-51).*

It should be noted that the two neighboring free blocks has to two pair of free two-bit values 01, 01 (See Table 1 in col. 7), then, after the step of coalescing (i.e. computing), a single

piece of memory is free having a free two-bit value 01, which is as result of logical union of 01 and 01.

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau and Forin at the time the invention was mage to modify the system of Armangau to include generating a summary map as a logical union of active maps as taught by Forin. One of ordinary skill in the art would be motivated to make this combination in order to representing an allocation state of the corresponding memory block in view of Forin, as doing so would give the added benefit of managing the memory blocks and tracks changes in allocation state via hierarchical bit map as taught by Forin *(col. 4, lines 24-30)*.

As per claim 26, Armangau teaches making write allocation decisions in said file system based on said summary map *(col. 2, lines 12-65)*.

As per claim 28, Armangau teaches said set of snapshots includes at least two snapshots; and said computing includes performing a bitwise logical operation on at least two said copies of earlier active maps included in said set of snapshots *(col. 2, lines 12-65; col. 14, line 62 to col. 15, line 18)*.

As per claim 29, Armangau teaches making write allocation decisions based on both current active map and said summary map *(col. 2, lines 12-65)*.

As per claim 30, Armangau teaches computing a combination of a current active map and said summary map; and making write allocation decisions based on a result of said computing *(col. 2, lines 12-65).*

As per claim 31, Armangau teaches for a selected portion of said summary map; identifying a set of snapshots created since recent update of said selected portion; and updating said selected portion based on only a most recent one of said snapshots *(col. 2, lines 12-65; col. 4, lines 13-47).*

As per claim 44, Armangau teaches making write allocation decisions in said file system based on said summary map *(col. 2, lines 12-65; col. 4, lines 13-47).*

As per claim 46, Armangau teaches said generating includes performing a bitwise logical operation on at least two said copies of earlier active maps included in said of persistent point-in-time images *(col. 2, lines 12-65; col. 4, lines 13-47).*

As per claim 47, Armangau teaches making write allocation decisions based on both a current active map and said summary map *(col. 2, lines 12-65; col. 4, lines 13-47).*

As per claim 48, Armangau teaches determining a combination of a current active map and said summary map; and making write allocation decisions based on a result of said computing *(col. 2, lines 12-65; col. 4, lines 13-47).*

4. Claims 32, 33, 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Armangau et al. (US Patent No. 6,434,681), in view of Goldstein et al. (US Patent No. 6,665,815).

As per claim 32, Armangau teaches a method comprising:

maintaining an active map *(i.e. data structure, col. 13, lines 25-65)* of information indicating in-use and free blocks associated with a file system *(i.e. a list of pointers to free tracks, col. 13, line 66 to col. 14, line 7);*

maintaining a set of snapshots *(i.e. snapshot copies, col. 13, lines 25-57)*, each snapshot representing a state *(i.e. the bit map indicates the modified state, col. 13, lines 58-65)* of said file system at a particular point in time *(i.e. date/time stamp when the snapshot copy was made, col. 18, lines 18-36); and*

maintaining a summary map *(Fig. 6)* based on an active map included in at least one of said snapshots *(i.e. snapshot track 0, snapshot track 1, Fig. 6) (col. 14, line 62 to col. 15, line 18).*

Armangau does not expressly teach receiving a request to delete a particular snapshot; and deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot, indicating said blocks is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot.

Goldstein teaches receiving a request to delete a particular snapshot; and deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot, indicating said blocks is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot *(col. 4, lines 41-51).*

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau and Goldstein at the time the invention was mage to modify the system of Armangau to receiving a request to delete a particular snapshot; and deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot, indicating said blocks is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot as taught by Goldstein. One of ordinary skill in the art would be motivated to make this combination in order to release blocks in the storage pool containing data unique to the base state snapshot and the first state snapshot in view of Goldstein *(col. 4, lines 41-51)*, as doing so would give the added benefit of the data volume is restored by restoring the base state data with data blocks contained in one or more succedent backups as taught by Goldstein *(col. 2, lines 40-58)*.

As per claim 33, Armangau teaches said indicating frees said block only when both said block is unused by said snapshot just prior to said particular snapshot; and said block is unused by said snapshot just after said particular snapshot *(col. 2, lines 12-65; col. 4, lines 13-47)*.

As per claim 34, Armangau teaches said snapshot just after said particular snapshot corresponds to an active file system *(col. 2, lines 12-65; col. 4, lines 13-47)*.

5. Claims 27, 40, 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Armangau et al. (US Patent No. 6,434,681), in view of Forin et al. (US Patent No. 6,175,900), and further in view of Ofek et al. (US Patent No. 6,092,066).

As to claims 27, 40, 45, Armangau and Forin do not expressly teach "summary map is computed using an inclusive OR operation".

Ofek teaches a summary map is computed using an inclusive OR operation *(i.e. performing a logical inclusive OR operation, col. 11, lines 4-12).*

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau, Forin and Ofek at the time the invention was mage to modify the system of Armangau and Forin to include a summary map is computed using an inclusive OR operation as taught by Ofek. One of ordinary skill in the art would be motivated to make this combination in order to indicate that the track in the local system is valid while the corresponding track in the remote system is no longer valid with respect to the data in the data storage sets in view of Ofek *(col. 11, lines 4-12)*, as doing so would give the added benefit to reestablish the second data storage facility as a mirror of the first data storage facility as taught by Ofek *(col. 3, line 47 to col. 4, line 5).*

6. Claim 41 is rejected under 35 U.S.C. 103(a) as being unpatentable over Armangau et al. (US Patent No. 6,434,681), in view of Goldstein et al. (US Patent No. 6,665,815), and further in view of Forin et al. (US Patent No. 6,175,900).

As per claim 41, Armangau teaches said summary map *(Fig. 6)* represents of at least two copies of an earlier active map included in at least two of said snapshots *(i.e. snapshot track 0, snapshot track 1, Fig. 6) (col. 14, line 62 to col. 15, line 18);*

Armangau and Goldstein do not expressly teach "a logical union".

Forin teaches a summary map *(i.e. a hierarchical bit-map 124, Fig. 8a)* represents as a logical union *(i.e. When two or more neighboring blocks are marked as free, they are automatically and immediately available for allocation as a single piece of memory, col. 9, lines 31-41)* of at least two copies of an earlier active maps *(i.e. the bit map, col. 7, lines 13-18) (col. 9, lines 9-51).*

It should be noted that the two neighboring free blocks has to two pair of free two-bit values 01, 01 (See Table 1 in col. 7), then, after the step of coalescing (i.e. computing), a single piece of memory is free having a free two-bit value 01, which is as result of logical union of 01 and 01.

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau, Goldstein and Forin at the time the invention was mage to modify the system of Armangau and Goldstein to include a summary map represents as a logical union of at least two copies of an earlier active maps as taught by Forin.

One of ordinary skill in the art would be motivated to make this combination in order to representing an allocation state of the corresponding memory block in view of Forin, as doing so would give the added benefit of managing the memory blocks and tracks changes in allocation state via hierarchical bit map as taught by Forin *(col. 4, lines 24-30).*

7. Claim 42 is rejected under 35 U.S.C. 103(a) as being unpatentable over Armangau et al. (US Patent No. 6,434,681), in view of Goldstein et al. (US Patent No. 6,665,815), and Forin et al. (US Patent No. 6,175,900), and further in view of Ofek et al. (US Patent No. 6,092,066).

As per claim 42, Armangau, Goldstein and Forin do not expressly teach "logical union is an inclusive OR operation".

Ofek teaches logical union is an inclusive OR operation *(i.e. performing a logical inclusive OR operation, col. 11, lines 4-12).*

It would have been obvious to one of ordinary skill of the art having the teaching of Armangau, Goldstein, Forin and Ofek at the time the invention was mage to modify the system of Armangau, Goldstein and Forin to include logical union is an inclusive OR operation as taught by Ofek. One of ordinary skill in the art would be motivated to make this combination in order to indicate that the track in the local system is valid while the corresponding track in the remote system is no longer valid with respect to the data in the data storage sets in view of Ofek *(col. 11, lines 4-12)*, as doing so would give the added benefit to reestablish the second data storage facility as a mirror of the first data storage facility as taught by Ofek *(col. 3, line 47 to col. 4, line 5).*

Response to Arguments

8. Applicant's arguments filed 11/17/06 have been fully considered but they are not persuasive.

Applicant argues that:

A. No logical operation.

Applicant argues that the difference between the Applicants invention and the Forin reference is:

Applicant's invention has combined/generated three of the four possible input states as: *free or free; free or in-used; in-used or free*.

The Forin reference has combined only one of four possible input state as: *free and free*.

The Examiner agrees with the Applicants about this point; however, *the claim languages do not reflect these limitations in order to distinguish these limitations with the Forin teachings.*

As detailed in the final rejection and the advisory action, to get the result as "free" from the inputs "free" or "in-used", the Forin applies the OR (i.e. logical union) operation as:

Outcome 1: 01 OR 01 = 01 (correct)

Outcome 2: 01 OR 11 = 11 (correct)

Outcome 3: 11 OR 01 = 11 (correct)

Outcome 4: 11 OR 11 = 11 (correct)

wherein the 01 is represented for the free state and 11 is represented for the in-used state.

B. The Examiner has focused only on "State 1" (01) and "State 2" (11), ignoring what would happen if the inputs to the logic operation include "State 3" (10) or "State 4" (00).

It is noted that since the claimed limitation recites "maintaining an active map of information indicating in-used blocks and free blocks", as detailed in the Advisory action, there are 4 possible outcomes, or all 4 possible inputs; it is hence only the inputs regarding the claimed limitations in-used and free blocks read on Forin reference. It would not be proper for the examiner to give words of the claim special meaning when no such special meaning has been

defined by the Applicant in the claim language as the Examiner's interpretation of the claim scope is consistent with term used.

B. Armangau does not teach or suggest "generating a summary map as a logical union of active maps".

Armangau teaches the step of generating a summary map as "*the bit map including a bit for each storage location of the production data set to indicate whether or not each storage location has been modified since the time when the snapshot copy is created*", col. 4, lines 13-47.

It is noted that, the step of generating a summary map is implied in the step of creating the snapshot copy as taught by Armangau.

Armangau does not expressly teach "generating a summary map as a logical union".

Forin teaches generating a summary map *(i.e. a hierarchical bit-map 124, Fig. 8a)* as a logical union *(i.e. When two or more neighboring blocks are marked as free, they are automatically and immediately available for allocation as a single piece of memory, col. 9, lines 31-41)* of active maps *(i.e. the bit map, col. 7, lines 13-18) (col. 9, lines 9-51).*

It should be noted that the two neighboring free blocks has two pair of free two-bit values 01, 01 (See Table 1 in col. 7), then, after the step of coalescing (i.e. computing), a single piece of memory is free having a free two-bit value 01, which is as result of logical union of 01 and 01.

Since the two references of Armangau and Forin directed to the same field as computer data storage and manage physical data memory using bitmap, it would have been obvious to a person having ordinary skill of the art at the time the invention was made to combine the invention of Armangau and Forin in arriving at the instant invention as set forth in claim 43. It is

within the scope of Armagau in view of Forin to employ "generating a summary map as a logical union" to manage allocation, deallocation, and reallocation of the memory blocks, and track the changes in allocation state via the hierarchical bitmap (as taught by Forin) and thus can be useful in maintaining in data storage of a data storage system a snapshot copy of a production data set including a multiplicity of storage locations in the data storage, where snapshot copy facility for a data storage system permitting continued host read/write access (as taught by Armangau). Applicant points out deficiencies of the Armangau and Forin reference singly. However, as is pointed out above in the 35 USC § 103 rejection argument, the Examiner has relied upon the **combine**d teaching of Armangau and Forin. It is the Examiner's opinion that the combination of the two references does yield the instant invention.

D. Armangau does not teach "maintaining an active map of information indicating in-used blocks and free-blocks associated with a file system".

Armangau teach an active map as "*a bit map for the data set*" wherein, *the bit map including a bit for each storage location of the production data set to indicate whether or not each storage location has been modified since the time when the snapshot copy is created.* The data processor is further programmed to respond to a request from the host processor for a write operation upon a storage location of the production data set by checking the bit in the bit map for the storage location of the production data set to determine whether or not the storage location of the production data set has been modified since the time when the snapshot copy was created, and upon finding that the storage location of the production data set has not been modified since the time when the snapshot copy was created, allocating a storage location to the snapshot copy, copying data from the storage location of the production data set to the allocated storage location of the snapshot copy, placing a pointer to the allocated storage location of the snapshot copy on the list of pointers allocated to the snapshot copy, changing the bit in the bit map for the storage location of the production data set, and after copying data from the storage location of the production

data set to the allocated storage location of the snapshot copy, performing the write operation upon the storage location of the production data set (col. 3, lines 23-55).

E. Armangau does not teach or suggest "generating a summary map as a logical union of active maps included in at least two of said persistent point-in-time".

Armangau teaches the step of generating a summary map as active map included in at least two of said persistent point-in-time as "*the bit map including a bit for each storage location of the production data set to indicate whether or not each storage location has been modified since the time when the snapshot copy is created*", col. 4, lines 13-47.

It is noted that the step of generating a summary map is implied in the step of creating the snapshot copy, and active map corresponds to the bitmap indicates the modified state (col. 13, lines 58-65), and at least two of said persistent point-in-time correspond to snapshot track 0, snapshot track 1 in Fig.6).

As detailed in the office action, Armangau does not expressly teach "a logical union".

Forin teaches generating a summary map *(i.e. a hierarchical bit-map 124, Fig. 8a)* as a logical union *(i.e. When two or more neighboring blocks are marked as free, they are automatically and immediately available for allocation as a single piece of memory, col. 9, lines 31-41)* of active maps *(i.e. the bit map, col. 7, lines 13-18) (col. 9, lines 9-51).*

It should be noted that the two neighboring free blocks has to two pair of free two-bit values 01, 01 (See Table 1 in col. 7), then, after the step of coalescing (i.e. computing), a single piece of memory is free having a free two-bit value 01, which is as result of logical union of 01 and 01.

Since the two references of Armangau and Forin are directed to the same field as computer data storage and manage physical data memory using bitmaps, it would have been obvious to a person having ordinary skill of the art at the time the invention was made to combine the invention of Armangau and Forin in arriving at the instant invention as set forth in claim 43. It is within the scope of Armagau in view of Forin to employ "generating a summary map as a logical union" to manage allocation, deallocation, and reallocation of the memory blocks, and track the changes in allocation state via the hierarchical bitmap (as taught by Forin) and thus can be useful in maintaining in data storage of a data storage system a snapshot copy of a production data set including a multiplicity of storage locations in the data storage, where snapshot copy facility for a data storage system permitting continued host read/write access (as taught by Armangau). Applicant points out deficiencies of the Armangau and Forin reference singly. However, as is pointed out above in the 35 USC § 103 rejection argument, the Examiner has relied upon the **combine**d teaching of Armangau and Forin. It is the Examiner's opinion that the combination of the two references does yield the instant invention.

F. Claim 32, The cited art also fails to disclose or suggest "deleting said particular snapshot, wherein said deleting involves for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot".

It is noted that claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Armangau et al. (US Patent No. 6,434,681), in view of Goldstein et al. (US Patent No. 6,665,815).

In this particular claim, the basis of the 35 USC 103 rejection is:

Armangau teaches a method comprising:

maintaining an active map *(i.e. data structure, col. 13, lines 25-65)* of information indicating in-use and free blocks associated with a file system *(i.e. a list of pointers to free tracks, col. 13, line 66 to col. 14, line 7);*

maintaining a set of snapshots *(i.e. snapshot copies, col. 13, lines 25-57)*, each snapshot representing a state *(i.e. the bit map indicates the modified state, col. 13, lines 58-65)* of said file system at a particular point in time *(i.e. date/time stamp when the snapshot copy was made, col. 18, lines 18-36); and*

maintaining a summary map *(Fig. 6)* based on an active map included in at least one of said snapshots *(i.e. snapshot track 0, snapshot track 1, Fig. 6) (col. 14, line 62 to col. 15, line 18).*

Armangau does not expressly teach receiving a request to delete a particular snapshot; and deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot, indicating said blocks is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot.

Goldstein teaches "deleting said particular snapshot, wherein said deleting involves for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot" as in the following paragraphs:

"Once the first succedent backup 131 has been created, *the first state snapshot 113 may be partially deleted*, as indicated by dashed lines in FIG. 5, and the base state snapshot 111 may be fully deleted. By "partial deletion" it means that the mapping metadata for first state snapshot 113 is not deleted. *With the deletion of the base*

state snapshot 111 and the partial deletion of the first state snapshot 113, it is possible to release blocks in the storage pool containing data unique to the base state snapshot 111 and the first state snapshot 113. In general, snapshots may be retained online for fast file recovery and/or deleted at a later time (col. 4, lines 41-51). As best seen in FIG. 6, a second succedent snapshot difference list 123 (S.sub.12) in state snapshots (i.e., S.sub.1.fwdarw.S.sub.2) is generated. A second succedent backup 133 (B.sub.12) is created from the second succedent snapshot difference list 123 by examining the snap disk metadata and copying from the second state snapshot 115 all the data blocks listed in the second succedent snapshot difference list 123 and copying the second succedent snapshot difference list 123 itself. Once the second succedent backup 133 has been made, *the second state snapshot 115 may be partially deleted, leaving at least the mapping metadata for the second state snapshot 115 (S.sub.2), and the first state snapshot 113 may be fully deleted"* (col. 4, lines 52-64).

The use of the knowledge would have been obvious to one ordinarily skilled in the art at the time of the invention to combine the teachings of Armangau with the teachings of Goldstein to include "delete a particular snapshot; and deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot, indicating said blocks is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot as taught by Goldstein" because both references teach methods for maintaining in data storage of a data storage system a snapshot copy of a production data set, and the incorporation of Goldstein in **the combined system** would have enhanced the performance of the system by efficiently releasing blocks in the storage pool containing data unique to the base state snapshot and the first state snapshot, as doing so would give the added benefit of having the data volume restored by restoring the base state data with data blocks contained in one or more succedent backups as taught by Goldstein *(col. 2, lines 40-58)*.

Applicant points out deficiencies of the Armangau and reference singly. However, as is pointed out above in the 35 USC § 103 rejection argument, the Examiner has relied upon the

combined teaching of Armangau and Goldstein. It is the Examiner's opinion that the combination of the two references does yield the instant invention.

Conclusion

9. This is a continuation of applicant's earlier Application No. 09932578. All claims are drawn to the same invention claimed in the earlier application and could have been finally rejected on the grounds and art of record in the next Office action if they had been entered in the earlier application. Accordingly, THIS ACTION IS MADE FINAL even though it is a first action in this case. See MPEP § 706.07(b). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no, however, event will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

/Miranda Le/

Miranda Le
February 02, 2007

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100